BigQuery and Prediction APIs
Getting more from your data with Google

Amit Agarwal
May 20th, 2010
View live notes and ask questions about this session on Google Wave
Overview

- Big Data - Challenging and Important
- Google has tools for deep data analysis
- Now you can use these tools
Overview

- Big Data - Challenging and Important
- Google has tools for deep data analysis
- Now you can use these tools
- Announcing two new APIs to get more from your data:
  1. BigQuery
  2. Prediction API
Benefits

- Built on Google technology
- Scalability
- Security
- Sharing
- Easy integration with Google App Engine, Google Spreadsheets, ....
Using Your Data with BigQuery & Prediction API

1. Upload
   - Upload your data to Google Storage

2. Process
   - Import to tables
   - Train a model

3. Act
   - Run queries
   - Make predictions

Your Data

BigQuery Prediction API

Your Apps
BigQuery
Interactive Analysis of Big Data
Siddartha Naidu
May 20th, 2010
Big Data is Challenging

Starts with **Scale**

Simplicity

Batch Jobs

Fast

BigQuery

Google
Many Use Cases ...

- Interactive Tools
- Spam
- Trends Detection
- Web Dashboards
- Network Optimization
Demo: Analyzing M-Lab

An open platform for advanced network research

http://www.measurementlab.net/

BigQuery

```
SELECT COUNT(*) FROM [mlab.v10];
```

Run
Demo: Exploring M-Lab

```sql
SELECT TOP(connection_spec.remote_hostname) Host, COUNT(*)
FROM [mlab.v10] WHERE
IS_EXPLICITLY_DEFINED(connection_spec.remote_hostname) AND
connection_spec.remote_hostname CONTAINS 'google';
```

<table>
<thead>
<tr>
<th>Host</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>216-239-45-4.google.com</td>
<td>52846352</td>
</tr>
<tr>
<td>216-239-44-65.google.com</td>
<td>107553</td>
</tr>
<tr>
<td>googlebroadband186.excellmedia.net.60.13</td>
<td>50181</td>
</tr>
<tr>
<td>googlebroadband187.excellmedia.net.60.13</td>
<td>29927</td>
</tr>
</tbody>
</table>
Key Capabilities of BigQuery

- **Scalable**: Billions of rows
- **Fast**: Response in seconds
- **Simple**: Queries in SQL
- **Web Service**
  - REST
  - JSON-RPC
  - Google App Scripts
Using Your Data with BigQuery

1. Upload
   Upload to Google Storage

2. Import
   Import data into a BigQuery Table
   - No need to define indices, keys, etc..

3. Query
   Execute queries via APIs
   - No provisioning machines or resources
Writing Queries

Compact subset of SQL
  o `SELECT ... FROM ...`
  WHERE ...
  GROUP BY ... ORDER BY ...
  LIMIT ...;

Common functions
  o Math, String, Time, ...

Statistical approximations
  o TOP
  o COUNT DISTINCT
API in a Minute

GET /bigquery/v1/tables/{table name}

GET /bigquery/v1/query?q={query}

Sample JSON Reply:
{
    "results": {
        "fields": [
            {
                "id": "COUNT(*)", "type": "uint64"
            }, ...
        ],
        "rows": [
            {
                "f": [{"v": "2949"}, ...],
                "f": [{"v": "5387"}, ...]
            }, ...
        ]
    }
}

Also supports JSON-RPC
Security and Privacy

Standard Google Authentication
- Client Login
- OAuth
- AuthSub

HTTPS support
- protects your credentials
- protects your data

Use Google Storage for Developers to manage access
Large Corpus Analysis

Wikimedia Revision History

BigQuery

```
SELECT TOP(title, 5), COUNT(*)
FROM [wikipedia]
WHERE wp_namespace = 0;
```

Wikimedia Revision history data from: http://download.wikimedia.org/enwiki/latest/enwiki-latest-pages-meta-history.xml.7z
Using BigQuery Shell

Python DB API 2.0 + B. Clapper's sqlcmd
http://www.clapper.org/software/python/sqlcmd/

```
<table>
<thead>
<tr>
<th>title</th>
<th>STRING NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>INT64 NULL</td>
</tr>
<tr>
<td>is_bot</td>
<td>BOOL NULL</td>
</tr>
<tr>
<td>comment</td>
<td>STRING NULL</td>
</tr>
<tr>
<td>num_characters</td>
<td>INT32 NULL</td>
</tr>
<tr>
<td>is_minor</td>
<td>BOOL NULL</td>
</tr>
</tbody>
</table>

? SELECT TOP(title, 5), COUNT(*) FROM [bigquery.test.001/tables/wikipedia]
> WHERE wp_namespace = 0;
Execution time: 10.953 seconds
5 rows

TOP(title, 5)            COUNT(*)
---------------------------
George W. Bush            43652
List of World Wrestling Entertainment employees 30572
Wikipedia                 29726
United States             27433
Michael Jackson           23245
```
BigQuery from a Spreadsheet

<table>
<thead>
<tr>
<th>Search Term</th>
<th><em>novalue</em></th>
<th>Num_Revs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No data
BigQuery from a Spreadsheet

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Num_Revs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>8755</td>
</tr>
<tr>
<td>Google search</td>
<td>4261</td>
</tr>
<tr>
<td>Google Earth</td>
<td>3874</td>
</tr>
<tr>
<td>Google Chrome</td>
<td>2687</td>
</tr>
<tr>
<td>Google Maps</td>
<td>2617</td>
</tr>
</tbody>
</table>

![Bar chart showing search term results](chart.png)
BigQuery Recap

- Interactive analysis of very large data sets
- Simple SQL query language
- APIs enable a variety of use cases
Prediction API
Machine learning as a web service

Max Lin, Gideon Mann
May 20th, 2010
Prediction API 101

- Google's sophisticated machine learning algorithms
- Available as an on-demand RESTful HTTP web service
- Train a model offline/asynchronously
- Predict results in real-time

"Tous pour un, un pour tous, c'est notre devise."
How does it work?

The Prediction API finds relevant **features** in the sample data during training.

| "english" | The quick brown fox jumped over the lazy dog. |
| "english" | To err is human, but to really foul things up you need a computer. |
| "spanish" | No hay mal que por bien no venga. |
| "spanish" | La tercera es la vencida. |

The Prediction API later searches for those **features** during prediction.

| ? | To be or not to be, that is the question. |
| ? | La fe mueve montañas. |
A virtually endless number of applications...

<table>
<thead>
<tr>
<th>Customer Sentiment</th>
<th>Transaction Risk</th>
<th>Species Identification</th>
<th>Message Routing</th>
<th>Diagnostics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Churn Prediction</td>
<td>Legal Docket Classification</td>
<td>Suspicious Activity</td>
<td>Work Roster Assignment</td>
<td>Inappropriate Content</td>
</tr>
<tr>
<td>Recommend Products</td>
<td>Political Bias</td>
<td>Uplift Marketing</td>
<td>Email Filtering</td>
<td>Career Counselling</td>
</tr>
</tbody>
</table>

... and many more ...
Three simple steps to use the Prediction API

1. Upload
   - Upload your training data to Google Storage
   - Use the API, gsutil or any compatible utility to upload your data to Google Storage

2. Train
   - Build a model from your data
   - prediction/v1/train/{}
     POST: a training request

3. Predict
   - Make new predictions
   - prediction/v1/query/{}
     GET: model info
     POST: a prediction request
Prediction API Demo
Automatically categorize and respond to emails by language

- **Customer**: ACME Corp, a multinational organization
- **Goal**: Respond to customer emails in their language
- **Data**: Many emails, tagged with their languages
- **Outcome**: Predict language and respond accordingly
Step 1: Upload
Upload your training data to Google Storage

- Training data: **outputs** and **input features**
- Data format: comma separated value format (CSV)

```
$ head -n 2 ${data}
"english","To err is human, but to really ..."
"spanish","No hay mal que por bien no venga."
```

Upload to Google Storage
```
$ gsutil cp ${data} gs://io10/${data}
```
Step 2: Train
Create a new model by training on data

To train a model:

POST prediction/v1/train/${data}

Training runs asynchronously. To see if it has finished:

GET prediction/v1/query/${data}

{"data": {
   "resource": {
      "data": "${data}'",
      "modelinfo": "estimated accuracy: ${acc}'"}}}}
Step 3: Predict
Apply the trained model to make predictions on new data

POST prediction/v1/query/${data}

{ data : {
  "instance" : {
    "input" : { "text" : [ "J'aime X! C'est le meilleur"]}
  }
}}
Step 3: Predict
Apply the trained model to make predictions on new data

POST prediction/v1/query/${data}

{ data : {
  "instance" : {
    "input" : { "text" : [
      "J'aime X! C'est le meilleur"
    ]}
  }
  "output" : {
    "output_label" : "french"}
}}
Step 3: Predict
Apply the trained model to make predictions on new data

import httplib

header = {"Content-Type" : "application/json"}  # ...put new data in JSON format in params variable
conn = httplib.HTTPConnection("www.googleapis.com")
conn.request("POST",
"/prediction/v1/query/${data}" , params, header)
print conn.getresponse()}
Prediction API Capabilities

Data
- Input Features: numeric or unstructured text
- Output: up to 100s of discrete categories

Training
- Many machine learning techniques
- Automatically selected
- Performed asynchronously

Access from many platforms:
- Web app from Google App Engine
- Apps Script (e.g. from Google Spreadsheet)
- Desktop app
Get the BigQuery & Prediction APIs

- Preview, opened to a limited number of developers

- To request access and get more information, go to:
  - http://code.google.com/apis/bigquery
  - http://code.google.com/apis/prediction
View live notes and ask questions about this session on Google Wave